Лабораторная работа №2

“Исследование сканера при анализе простых языковых конструкций”

1 ПОСТАНОВКА ЗАДАЧИ

Вариант 33 – 3, 4, 14

Таблица 1 – Типы идентификаторов

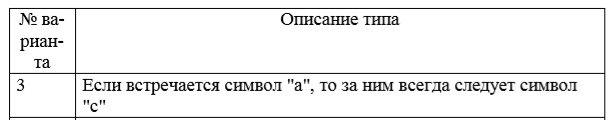


Таблица 2 – Типы констант

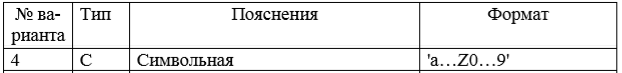


Таблица 3 – Фрагмент программы для анализа



2 ХОД РАБОТЫ

1. Назначим коды лексемам:

FUNC – 100

FINAL – 200

идентификатор, <iden> - 300

константа, <data> - 400

: – 501

\ – 502

! – 503

{ – 504

} – 505

( – 506

) – 507

:= – 508

&& – 509

!! – 510

2. Построение минимального детерминированного конечного автомата:

1) Служебные слова являются самоопределяющимися цепочками, составленными из литер {F, U, N, C, I, A, L} – FUNC, FINAL.

2) Эскизно идентификатор (переменная) будет выглядеть так:

*( Б* ∪ *Ц* ∪ *AC ) { Б* ∪ *Ц* ∪ *AC } ,*

где *Б*- любая латинская буква кроме A; *Ц*– любая десятичная цифра

3) Символьная константа описывается следующим образом:

*‘( Б* ∪ *Ц* ∪ *A ) { Б* ∪ *Ц* ∪ *A }’ ,*

где *Б*– любая латинская буква кроме A; *Ц*– любая десятичная цифра

4) Множество букв Б = {F, U, N, C, I, L, δ}

где δ – буква латинского алфавита, не совпадающая по начертанию с F, U, N, C, I, L.

5) Множество цифр Ц = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

6) Множество «не буква, не цифра» L1 ={:, \, !, {, }, (, ), :=, &&, !!} ∪ L3

7) Множество L2 = L1 – текущая литера не принадлежит к образующим константу или литерал.

8) L3 – литера, не принадлежащая алфавиту конечного автомата.

9) FUNCL1 – первое служебное слово.

10) FINALL1 – второе служебное слово.

11) (F ∪ U ∪ N ∪ C ∪ I ∪ L ∪ δ ∪ Ц ∪ AC) {F ∪ U ∪ N ∪ C ∪ I ∪ L ∪ δ ∪ Ц ∪ AC} L1 – переменная (iden).

12) “{F ∪ U ∪ N ∪ C ∪ I ∪ A ∪ L ∪ δ ∪ Ц } (F ∪ U ∪ N ∪ C ∪ I ∪ A ∪ L ∪ δ ∪ Ц)” L2 – константа.

13) : ∪ \ ∪ ! ∪ { ∪ } ∪ ( ∪ ) ∪ := ∪ && ∪ !! ∪ L3 – однолитерные и двулитерные разделители

3. Разметка, приведение к ДКА и минимизация в exсel.

4. Код программы:

import java.io.FileReader;

import java.util.Scanner;

public class Main {

private static int[][] matrix;

private static Scanner scanner;

private static boolean flag = false;

private static int temp\_ch;

public static void main(String[] args) {

try{

int i, n;

FileReader fr = new FileReader("data.txt");

matrix = readMatrix(matrix); // управляющая таблица

i=0; // номер состояния КА

n=0; // число правильных слов

String word = new String();

int ch; // текущая литера

ch = fr.read();

while (ch != -1) {

if (Character.isLetter(ch)){

switch(ch) {

case 'F':

i = matrix[0][i];

word += (char) ch;

break;

case 'U':

i = matrix[1][i];

word += (char) ch;

break;

case 'N':

i = matrix[2][i];

word += (char) ch;

break;

case 'C':

i = matrix[3][i];

word += (char) ch;

break;

case 'I':

i = matrix[4][i];

word += (char) ch;

break;

case 'A':

i = matrix[5][i];

word += (char) ch;

break;

case 'L':

i = matrix[6][i];

word += (char) ch;

break;

default:

i = matrix[7][i];

break;

}

}

else if (Character.isDigit(ch)) {

i = matrix[8][i];

}

else {

switch (ch) {

case ':':

i = matrix[9][i];

word += (char) ch;

break;

case '\\':

i = matrix[10][i];

word += (char) ch;

break;

case '!':

i = matrix[11][i];

word += (char) ch;

break;

case '{':

i = matrix[12][i];

word += (char) ch;

break;

case '}':

i = matrix[13][i];

word += (char) ch;

break;

case '(':

i = matrix[14][i];

word += (char) ch;

break;

case ')':

i = matrix[15][i];

word += (char) ch;

break;

case '=':

i = matrix[16][i];

word += (char) ch;

break;

case '&':

i = matrix[17][i];

word += (char) ch;

break;

case ' ':

i = matrix[20][i];

word += (char) ch;

break;

case '"':

i = matrix[19][i];

word += (char) ch;

break;

default:

i = matrix[18][i];

word += (char) ch;

break;

}

}

if ((i > 99)&&(i < 500)) {

/\* Анализ кода состояния \*/

switch(i){

case 100:

System.out.println("Служебное слово 'FUNC'. Cостояние = "+i);

break;

case 200:

System.out.println("Служебное слово 'FINAL'. Cостояние = "+i);

break;

case 300:

System.out.println("Завершился идентификатор. Cостояние = "+i);

break;

case 400:

System.out.println("Завершилась константа. Cостояние = "+i);

break;

}

i=0;

if (ch == ':' || ch == '!' || ch == '&') flag = true;

}

else if (i > 500 && i < 800) {

switch(i) {

case 501:

System.out.println("Разделитель ':'. Cостояние = " + i);

break;

case 502:

System.out.println("Разделитель '\\'. Cостояние = " + i);

break;

case 503:

System.out.println("Разделитель '!'. Cостояние = " + i);

break;

case 504:

System.out.println("Разделитель '{'. Cостояние = " + i);

break;

case 505:

System.out.println("Разделитель '}'. Cостояние = " + i);

break;

case 506:

System.out.println("Разделитель '('. Cостояние = " + i);

break;

case 507:

System.out.println("Разделитель ')'. Cостояние = " + i);

break;

case 508:

System.out.println("Разделитель ':='. Cостояние = " + i);

break;

case 509:

System.out.println("Разделитель '&&'. Cостояние = " + i);

break;

case 510:

System.out.println("Разделитель '!!'. Cостояние = " + i);

break;

default:

System.out.println("Ошибка "+i);

}

i=0;

}

else if (i > 800) {

switch(i) {

case 801:

System.out.println("Неправильное начало");

break;

case 802:

System.out.println("Ошибка в служебном слове");

break;

case 803:

System.out.println("Ошибка в написании имени переменной");

case 804:

System.out.println("Ошибочная константа");

case 805:

System.out.println("Ошибка");

default:

System.out.println("Неизвестная входная литера");

}

i=0;

ch = fr.read();

}

if (!flag){

ch = fr.read();

}else{

flag = false;

}

}

}

catch (Exception e){

e.printStackTrace();

}

}

public static int[][] readMatrix(int mas[][]) {

try{

scanner = new Scanner(new FileReader("matrix.txt"));

int n = scanner.nextInt();

int m = scanner.nextInt();

if (scanner.hasNextInt()){

mas = new int [n][m];

for (int i=0; i<n; i++) {

for (int j = 0; j < m; j++ ) {

mas[i][j] = scanner.nextInt();

}

}

}

} catch (Exception e){

e.printStackTrace();

}

return mas;

}

public static void Print(int[][] matrix) {

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

System.out.println();

}

}

5. Результаты выполнения

Входная последовательность: FUNC(x)\ { X:=(x&&2)\ P:=(x!!9)\ } FINAL

Результат: Верно

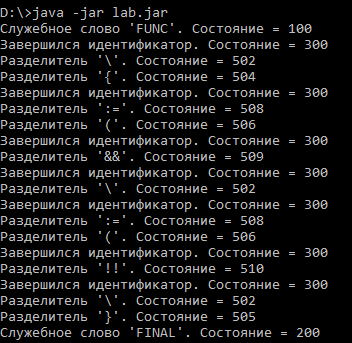


Рисунок 1 – Положительный исход

Входная последовательность: FUNC(x)\ { X:="123"\ } FINAL

Результат: Верно

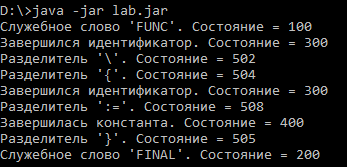


Рисунок 2 – Положительный исход

Входная последовательность: FUNC(x)\ { X:=123\ } FINAL

Результат: Не верно – отсутствуют кавычки

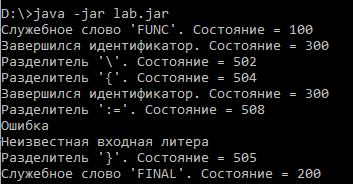


Рисунок 3 – Отрицательный исход

Входная последовательность: FUNC(x)\ { \*:="test" } FINAL

Результат: Не верно – некорректный идентификатор

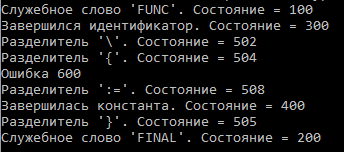


Рисунок 4 – Отрицательный исход